UltimateSky

April 6, 2016

Contents

1 Introduction

UltimateSky is a configurable seasonal skybox shader that intends to model the sky accurately in all its phases. It has a day and night cycle which it models using a calendar and geolocation. The configuration options are defined in a scriptable object and include calendar and sky composition. In addition it uses a cubemap for the star sky with accurate star positions. It is easy to create new configuration objects and the skybox will instantly respond if the config is switched. In addition the cubemap for the stars can be generated from a catalogue of known stars, or be replaced with fictional and/or random.

The single UltimateSky prefab required in the scene contains two components, the main UtimateSky component and the Calendar component. The main component is used to set the geolocation, overcast as well as other important values, and it updates the shader material. The Calendar component sets and tracks the date and time.

The skybox can also render two layers of clouds from seamless noise textures, for a simple weather effect but its should be noted that in itself this is not intended to be a fullblown weather solution.

In summary, its an easy to use dynamic day and night skybox shader, easy to control control from scripts or left to run on its own. The system is very simple, but has gone trough many iterations to ensure a good quality.

2 Requirements

The package does not depend on any Standard Assets.

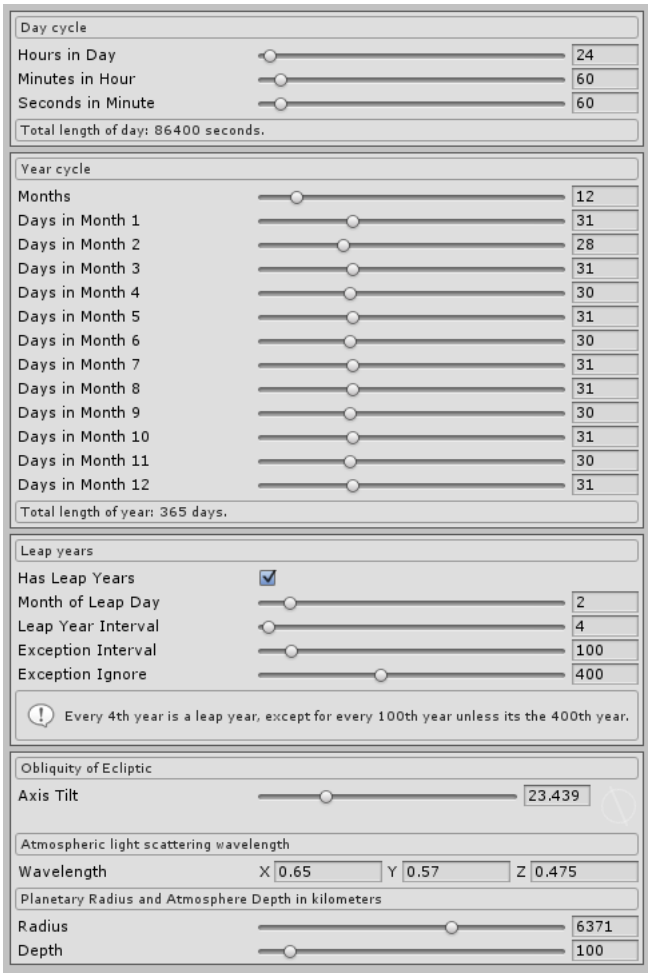
The shader is using Shader Model 3.

There are two extra Demo scenes that depend on Environments and Characters packages from the Standard Assets.

3 The Prefabs

3.1 The UltimateSkyConfiguration Data Asset

To create a new configuration select from the Menu "Assets/Create/UltimateSky/Configuration". You will get a new scriptable object with an inspector like the one in the following image.



Filling out the calendar should be pretty obvious so ill skip explaining that. The last section on the other hand could be explained a bit.

3.1.1 Obliquity of Ecliptic

This is the axial tilt of the planet in reference to its orbital plane. If this is zero, then there will be no seasons. If it is extreme, then the sun will stay above or below the horizon for long periods.

3.1.2 Atmospheric Light Scattering

This will produce the color of the sky. I don't know the physics to help you figure out what you want, just experiment until you get a desired effect.

3.1.3 Planetary Radius and Atmosphere Depth

These values will have an effect on the strength and depth of the color. Some values may produce undesired artifacts below the horizon, change the values or hide it under terrain if it becomes an issue.

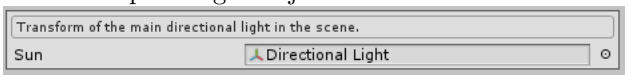
3.2 The UltimateSky Prefab

The prefab has two components, UltimateSky and UltimateSkyCalendar. Here follows a breakdown of each.

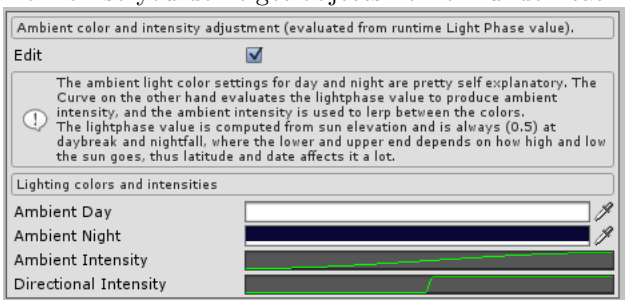
3.2.1 UltimateSky Component

First there is a slot for setting the active configuration data asset. This can also be changed on the fly and the skybox will respond immediately with no delay.

Next there is the slot for setting the directional light that will act as the sun. Its best to keep the light object in the center of the scene.

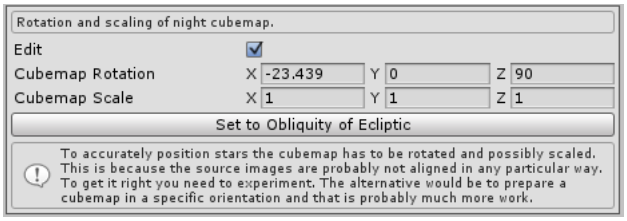


Then there are the settings for light. First of all you define the ambient light colors for day and night. Then you supply a curve that will define the ambient light intensity throughout the cycle. Finally there is a curve for the directional light intensity. This will help by cutting the main light once the sun goes below the horizon so you don't get objects lit from underneath the terrain.

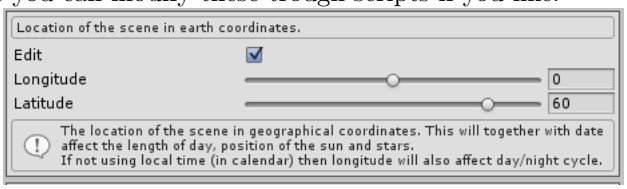


Next is the color to blend on the skybox below the horizon.

The cubemap for the stars is set directly in the material (more on the material later), in here you can set some transformations on it. You can just click to set it to the obliquity of the ecliptic and that will work well with the cubemap provided or generated by the utility provided. But if you have a custom cubemap and you need to align it properly you can do it here.

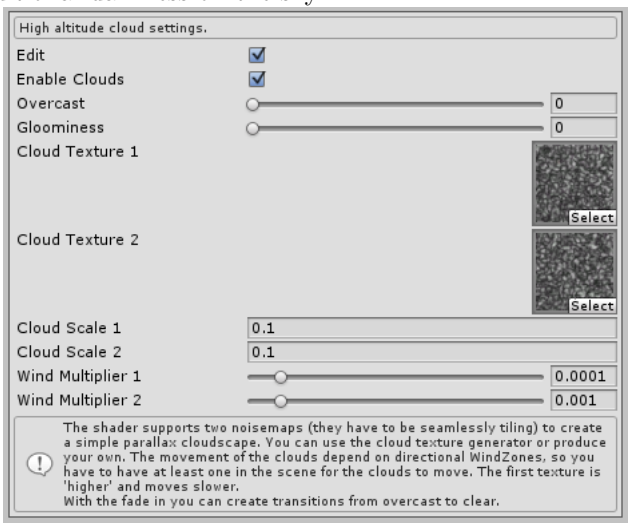


The longitude and latitude is set next. These will remain fixed by default but you can modify these trough scripts if you like.

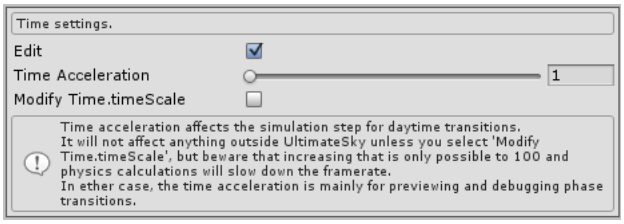


Next we have settings for cloud layers. So we support two textures that are affected by directional windzones on the scene. They have to be seamless to work flawlessly. By reducing the scale value the textures will stretch out more. Also by providing two different wind multipliers you get a parallax effect on the layers.

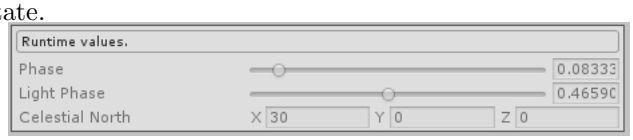
The overcast is what fades the clouds in or out and the gloominess adds additional darkness on the sky.



The time acceleration is useful mostly to debug or to create timelapses. If the modify timescale is enabled, then the actual time in unity will run faster is absolutely needed (there is a practical limit to this however and things might bet a bit buggy).

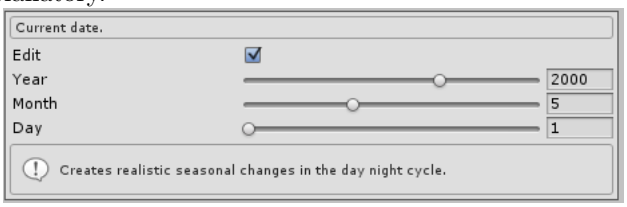


Finally there is a box showing some runtime values. The phase is the linear progression throughout the cycle, whereas the LightPhase has more to do with Sun elevation. Celestial north points to the point in the sky around which things rotate.

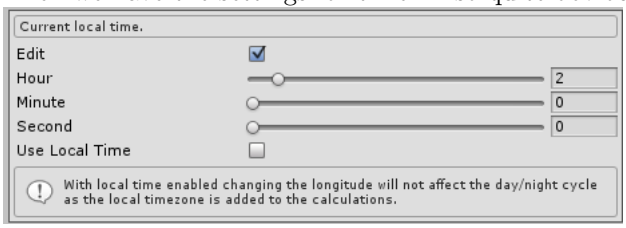


3.2.2 UltimateSkyCalendar Component

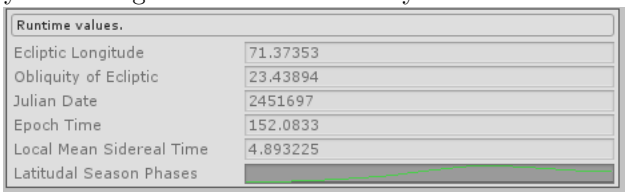
In the calendar component we have the settings for date. Should be pretty self explanatory.



Then we have the settings for time. Also quite obvious what this does.



The runtime debug box shows the values that has been calculated. These can of course be read trough scripts, if you find them useful for some extension. The latitudinal season phase is not used by this system but has been calculated because it can also be useful. It ranges from 0-3 (winter to summer). You use it by evaluating the latitude as the key.



4 The Skybox Material

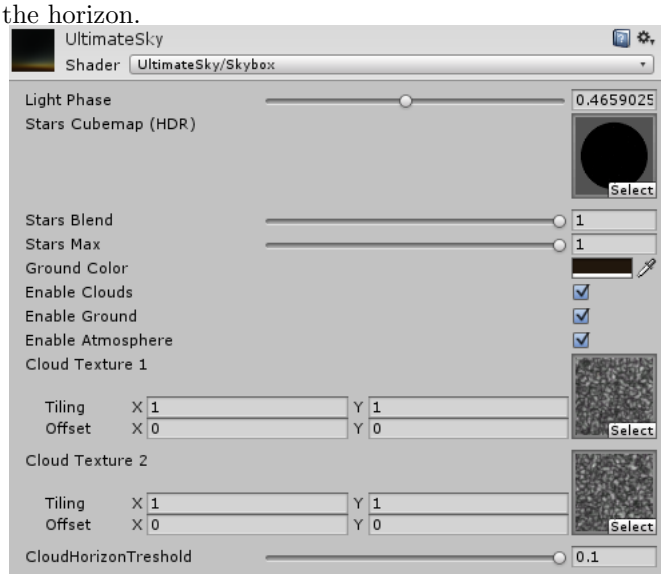
This is the material that you should set to your skybox in the Lighting settings window.

In the material inspector of the skybox (UltimateSky.material) you set the cubemap that is used for the night sky. Some of the other properties are controlled by the prefab, but some others you can adjust only here and ill explain them now.

Stars blend will tell how much of the cubemap will be blended in. Stars max will do the same but by performing an cutout test. They produce almost the same effect but in some cases the other will do better (depending on effect). By default both are at max, but if you use a custom cubemap you might like to play around with these too.

Enable ground and enable atmosphere are only available here, and by disabling both you get a complete starscape all around.

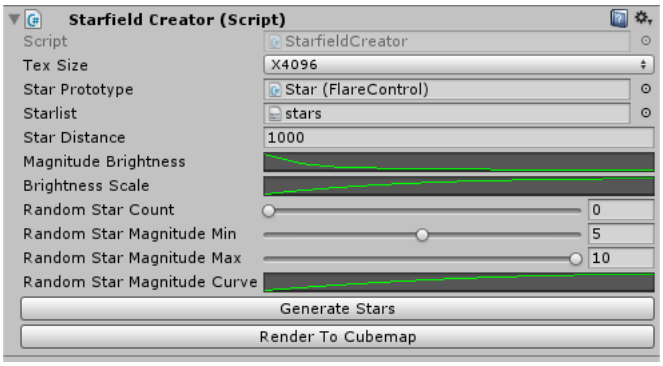
Cloud horizon threshold can be used to tweak a bit how the clouds terminate at the horizon.



5 Generating a Cubemap

You can of course generate your starfield in any way you want, but i have supplied a simple utility scene that also does it. It is located in "Assets/UltimateSky/Utils/Starfield".

Its all set up and ready to use. It contains an gameobject called StarfieldCreator and the cubemap is created by the controls in its inspector.



First there is the desired size of the cubemap, the higher the better. When you create a new cubemap asset Unity might by default be importing it at a lower resolution so you should check the import settings of the newly created cubemap.

Then there is the prototype for the stars, im using flares to do it.

Then there is the star catalogue. This is an csv file that contains some 1600 of the brightest stars. You can create your own list if you want.

You can play around with the distance, magnitude and brightness values but changing them too much might just break it. So try to keep a backup of it.

The last fields allows you to saturate the sky with more random stars. How many random stars you can create depends on your system, and trying to create too many might take long or you might run out of memory. So save everything before running this util.

6 Step by Step Guide

6.1 Directional Light

Make sure you have a directional light on the scene and that its transform position is exactly [0,0,0].

6.2 Skybox Material

1) Locate the skybox material "Assets/UltimateSky/Materials/UltimateSky.mat". The material is all set to go, but if you want you can change the nightsky cubemap associated.

2) Open "Window/Lighting" and select the "Scene" tab.

3) Drag the material into the skybox material slot.

4) While here, also drag the main Directional Light into its corresponding slot. Obviously you need one for this to work.

5) Select the ambient source; can be Skybox or Color. If you select color, then UltimateSky will transition between two colors and intensities for day and night.

6.3 UltimateSky Prefab

1) Locate the skybox controller at "Assets/UltimateSky/Prefabs/UltimateSky.prefab".

2) Drag the prefab to the scene.

6.3.1 The Main Component

1) Assign a configuration object to the instance. For example pick the "Earth-Sky".

2) Assign the main Directional Light to the Sun slot.

3) Assign the cloud noise textures or alternatively disable the clouds by making sure the "Enable Clouds" checkbox is ticked off. There are a few seamless nose textures you can use for the clouds at "Assets/UltimateSky/Noise".

4) Go to time settings and drag the "Acceleration" to maximum and hit play.

5) You probably noticed that the clouds didn't move. Add a directional Windzone to the scene and hit play again. The clouds will more quite rapidly when there is time acceleration, so drop that down to 1 (please notice that the minimum is 0, which effectively stops all progression). You should now have everything set up as intended.

6.3.2 The Calendar Component

Once you have all previous steps completed and the skybox working as intended you will most likely want to start playing around with the calendar which is a component on the same prefab you already added to the scene.

Another thing you may want to play around with is the longitude, latitude and overcast strength (part of main component).

6.4 The Demo Scene

To get the best idea of how things work i recommend opening the Demo scene which as a small HUD component where you can adjust all the important values. The script for that HUD component also clearly demonstrates how the skybox can be controlled from scripts.